

Claims:

1. A method for the automatic installation and configuration of software components (SW) in a computer
5 network (1) which comprises a plurality of client computers (2) and at least one network resource (RES) of installable software components, characterised by the steps:

a) provision of a framework (FW) on the network resource (RES) which comprises a rule package (RP) for each of the
10 installable software components (SW) of the network resource (RES) and a list (L) of rule packages (RP) to be run, but not the software components (SW) themselves,

wherein at least one of the rule packages (RP) comprises a routine (4) for loading its software component (SW) from the
15 network resource (RES) and installing it on a client computer (2) and at least this or one of the other rule packages (RP) comprises a routine (5) for configuring its software component (SW) installed on a client computer,

b) transferring the entire framework (FW) to a client
20 computer (2); and

c) running the list (L) of rule packages (RP) with installation routines (4) to be run on the client computer (2), calling their installation routines (4), and again
25 running the list (L) of rule packages with configuration routines (5) to be run on the client computer (2), calling their configuration routines (5),

wherein at least step c) is triggered by a local event (16-19) on the particular client computer (2).

2. A method according to claim 1, characterised in that
30 step c) is triggered by a system startup or shutdown, system lock or share, user logon or logoff, network logon or logoff, program startup or shutdown, connection or disconnection of hardware or by a timer.

3. A method according to claim 1 or claim 2, in which
35 successful installation of a software component on a client

computer may have as a prerequisite the presence or absence, configuration or deconfiguration of another software component, characterised in that,

in step a), the framework (FW) comprises a detector (DET)
5 for each possible prerequisite and at least one of the rule packages (RP) comprises a routine (4') for deinstalling its software component from a client computer (2) and at least this or one of other rule packages (RP) comprises a routine (5') for undoing (deconfiguring) the configuration of its
10 software component (SW) on a client computer (2), and,

in step c), if in the course of a rule package (RP) it is established by means of a detector (DET) that the presence or absence, configuration or deconfiguration of another software component (SW) is necessary, the installation or
15 deinstallation routine, configuration or deconfiguration routine (4, 4', 5, 5') of the rule package (RP) assigned to this other software component (SW) is called.

4. A method according to any one of claims 1 to 3, characterised in that the framework (FW) also comprises
20 detectors (DETHW, DETBS1) for a client computer's (2) hardware or operating system and, in the course of a routine (4, 4', 5, 5'), it is verified by means of such a detector whether the client computer (2) is suitable for the particular installation, deinstallation, configuration or deconfiguration
25 of the software component (SW).

5. A method according to any one of claims 1 to 4, characterised in that, in the course of a routine (4, 4', 5, 5'), it is checked in advance whether the particular
30 installation, deinstallation, configuration or deconfiguration of the software component (SW) has already taken place on the client computer (2) and, if so, the routine is immediately terminated.

6. A method according to any one of claims 1 to 5, characterised in that step b) and/or step c) is also triggered

by a remote event on the network resource, preferably the transmission of a group or broadcast message.

7. A rule package which is executable on an operating system of a client computer (2) for the automatic installation and configuration of software components (SW), which are available on a network resource (RES), on the client computer (2), characterised in that the rule package (RP) comprises a reference (RES_A) to a software component on the network resource (RES) and comprises at least one of the following four routines: a routine (4) for installing this software component (SW) on the client computer (2), a routine (4') for deinstalling this software component (SW) from the client computer (2), a routine (5) for configuring said software component (SW) installed on the client computer (2), and a routine (5') for undoing (deconfiguring) the configuration of this software component (SW) installed on the client computer (2), wherein each routine (4, 4', 5, 5'), if it establishes a presence or absence requirement of another software component (SW), branches to the installation or deinstallation routine (4, 4') of another rule package (RP) assigned to this other software component (SW).

8. A rule package according to claim 7, characterised in that it comprises a reference (DET_{HW}, DET_{BS1}) to a client computer's (2) specific hardware and/or operating system and, by means of this reference, verifies whether the client computer (2) is suitable for the particular installation, deinstallation, configuration or deconfiguration of the software component (SW).

9. A rule package according to claim 7 or claim 8, characterised in that it verifies whether the particular installation, deinstallation, configuration or deconfiguration of the software component (SW) on the client computer (2) has already occurred and, if so, terminates its execution.

10. A rule package according to any one of claims 7 to 9, characterised in that it contains at least one trigger

reference (TRIG) to a local event (16-19) on the client computer (2), wherein the trigger reference (TRIG) assigns at least one of the routines (4, 4', 5, 5') of the rule package to this event.

5 11. A rule package according to any one of claims 7 to 10, characterised in that it further contains at least one trigger reference (TRIG) to a remote event on the network resource, wherein the trigger reference (TRIG) assigns at least one of the routines (4, 4', 5, 5') of the rule package
10 to this event.

12. A rule package according to any one of claims 7 to 11, characterised in that it may be put in an inactive state in which only its deinstallation and deconfiguration routines (4', 5') can be called.

15 13. A computer which is programmed with at least one rule package according to any one of claims 7 to 12.

14. A framework which may be provided on a network resource (RES) in a computer network (1) for a plurality of client computers (2) for the automatic installation and
20 configuration on the client computers (2) of software components (SW) available on the network resource (RES), wherein successful installation of a software component (SW) may have as a prerequisite the presence or absence of another software component (SW), characterised in that the framework
25 (FW) comprises a set of rule packages (RP) according to any one of claims 7 to 12, a set of detectors (DET) for each possible prerequisite, and a list (L) of rule packages (RP) to be run on the client computers (2).

15. A framework according to claim 14 in conjunction with
30 a rule package according to claim 8, characterised in that the framework (FW) also comprises detectors (DETHW, DETBS1) for a client computer's (2) hardware or operating system and provides the rule packages (RP) for the stated verification.

16. A computer which is programmed with a framework
35 according to claim 14 or claim 15.

17. A machine-readable data storage medium which is programmed with a framework according to claim 14 or claim 15.

18. A client program which is executable on a client computer (2) for the automatic installation and configuration
5 of software components (SW), which are available on a network resource (RES), on the client computer (2), characterised in that it receives and stores a framework (FW) according to claim 14 or claim 15, in a first pass runs the list (L) of rule packages (RP) to be run, calling their installation
10 routines (4), and in a second pass runs the list (L) of rule packages (RP) to be run, calling their configuration routines (5).

19. A client program according to claim 18, characterised in that it comprises a local database (DB) which contains a
15 list (7) of rule packages (RP) with installation routines (4) which have run successfully and a list (8) of rule packages (RP) with configuration routines (5) which have run successfully.

20. A client program according to claim 19, characterised
20 in that it compares the rule packages (RP) entered in the lists (7, 8) with the rule packages (RP) contained in the framework (FW) and, for those rule packages (RP) which do not appear in the framework (FW), runs their deconfiguration routines (5') in a first pass and their deinstallation
25 routines (4') in a second pass.

21. A client program according to any one of claims 18 to 20 in conjunction with a rule package according to claim 10, characterised in that it monitors the occurrence of a local event (16-19) on the client computer (2), preferably a system
30 startup or shutdown, system lock or share, user logon or logoff, network logon or logoff, program startup or shutdown, connection or disconnection of hardware or response of a timer, and calls the corresponding rule package (RP) routine (4, 4', 5, 5') which is assigned via the trigger reference
35 (TRIG) to said event.

22. A client program according to any one of claims 18 to 20 in conjunction with a rule package according to claim 11, characterised in that it further monitors the occurrence of a remote event on the network resource, preferably the
5 transmission of a group or broadcast message, and calls the corresponding rule package (RP) routine (4, 4', 5, 5') which is assigned via the trigger reference (TRIG) to this event.

23. A client program according to any one of claims 18 to 22, characterised in that it comprises a transaction system
10 for each system-modifying component, in particular for the rule packages (RP).

24. A computer which is programmed with a client program according to any one of claims 18 to 23.

25. A computer program implementing a method according to any
15 one of claims 1 to 6.